Session 120

Carbon Controls and Appearance II

Guy Fullerton
High Level Toolbox

Overview

- Control manager
- Dock integration
- Control Carbon events
- Data browser
- Help tags
- Fine tuning

Control Manager Features

- Control animation
- New control definitions
- Control sizes
- Moving controls between windows

Aqua Widget Animation

- Automatic animation:
  - Default buttons, progress bars, chasing arrows, edit text
- IdleControls() is unnecessary under Aqua
- Animation starts after a normal draw
- Stop/Start animation via SetControlData() and kControlProgressBarAnimatingTag or kControlChasingArrowsAnimatingTag

Animating Push Buttons

- Must be marked as default
- SetWindowDefaultButton()
- SetDialogDefaultItem()
- SetControlData() and kControlPushButtonDefaultTag

New Widgets

- Rounded bevel buttons
- Round button
- Edit Unicode text
- Relevance bar
- Disclosure button
- Four-sided tabs
- Appearance Manager support too

Control Sizes

Control Sizes

- Normal, small, large, and automatic
- Normal: platinum-sized
- Automatic: size determined by control bounds
- Follow Human Interface Guidelines!

Control Sizes

- SetControlData() and kControlSizeTag
- kControlSizeNormal/Small/Large/Auto
- Not all controls support all sizes
- Mac OS X only

Embedding Is Mandatory!

- Control Embedding is automatic on Mac OS X
- You must turn it on manually on Mac OS 9
  - CreateRootControl()
- Hit testing vs. draw order
- Enables future Control Mgr enhancements

Moving Controls Between Windows

- Developer request
  - Drag and Drop of interface elements
  - Save memory and time
- Carbon doesn't let you hack the control list
- Use EmbedControl() to move a ControlRef
- Can't move the Root Control

SendControlMessage()

- Don't send standard CDEF messages directly!
  - CarbonEvents are the native messaging model
- Use normal Control Manager APIs instead
- Sending messages to Custom CDEFs is still okay

The Future

- Drawing with Quartz
- Control drawing modes

Drawing With Quartz

- Appearance primitives
  - May pass in a CGContextRef
  - Draw animating push button
- Control Definitions will receive a CGContextRef
  - Faster and more efficient

Control Drawing Modes

- Enables transparent sheets in Carbon
- Improved compositing
- Batched drawing for better performance
- Predictable drawing for developer sanity

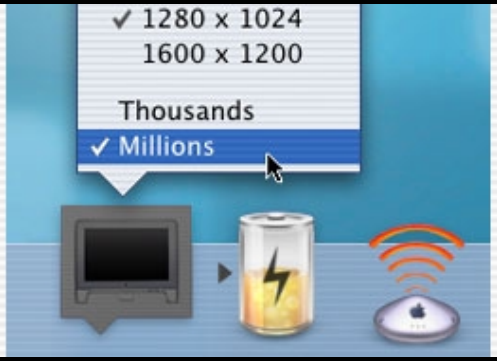
Animating Application Icons

- Draw with QuickDraw:
  - Begin/EndQDContextForApplicationDockTile()
- Draw with Quartz:
  - Begin/EndCGContextForApplicationDockTile()
  - SetApplicationDockTileImage()
- Badging with Quartz:
  - Overlay/RestoreApplicationDockTileImage()
- Header documentation in MacApplication.h

Minimized Window Icons

- Updating based on new window content:
  - UpdateCollapsedWindowDockTile()
- Custom tile drawing:
  - CreateQDContextForCollapsedWindowDockTile()
  - ReleaseQDContextForCollapsedWindowDockTile()
- Header documentation in MacWindows.h

Demo



David McLeod
Demo Boy II

Dock Size and Location

- Avoid zooming behind the Dock!
  - GetAvailableWindowPositioningBounds()
- Provide users with intelligent experience

Control Carbon Events

- Easier way to write Control Def Procs
- Allow functionality augmentation
- Allow performance enhancements
- kEventClassControl
- CarbonEvents.h

Control Creation

- RegisterToolboxObjectClass()
  - List of CarbonEvents you desire
  - EventHandler procedure
- CreateCustomControl()
  - ControlDefSpec with your ToolboxObjectClassRef
  - Initialization Collection
- RegisterControlDefinition()
  - Facilitates GetNewControl()

Lifecycle

- kEventControlInitialize
  - Receives initialization Collection in kEventParamInitCollection
  - Must return control features in kEventParamControlFeatures
- kEventControlDispose

Client Expectations

- kEventControlGetOptimalBounds
  - Sent from GetControlBestRect()
- kEventControlGetPartRegion
  - Sent from GetControlRegion()
  - kControlStructure/ContentMetaPart
- kEventControlGet/SetData
  - Sent from Get/SetControlData()

Rendering

- kEventControlDraw
  - kEventParamGrafPort contains the drawing port
- kEventControlApplyBackground and kEventControlApplyTextColor
  - Required for many embedders

Mouse Tracking

- kEventControlHitTest
  - Mouse point comes in…
  - …part code goes out
- Basic mouse tracking happens automatically
- kEventControlTrack
  - Total customization

Focusing

- kEventControlSetFocusPart
  - kEventParamControlPart has the part to focus:
    - kControlFocusNoPart: turn off the focus
    - kControlFocusNext/PrevPart: advance or reverse the focus
    - Other part code: focus that specific part
  - On exit, store the actual focused part in kEventParamControlPart
- kEventControlGetFocusPart
  - Pass back your currently focused part in kEventParamControlPart

Keyboard Input

- Events are routed directly to the user focus
- kEventClassTextInput and kEventTextInputUnicodeForKeyEvent
- kEventParamTextInputSendText
  - Contains the Unicode character stream

More User Events

- kEventControlActivate/Deactivate
  - Change your internal representation
  - No need to redraw
- kEventControlSetCursor
  - Sent from HandleControlSetCursor()
- kEventControlContextualMenuClick
  - Sent from HandleControlContextualMenuClick()

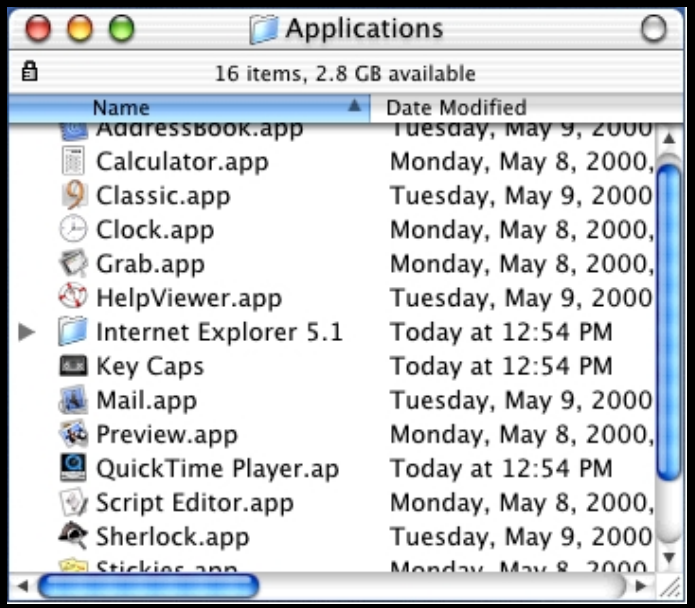
State Change Notifications

- kEventControlValueFieldChanged
- kEventControlAddedSubControl
- kEventControlRemovingSubControl
- kEventControlBoundsChanged
- kEventControlOwningWindowChanged

Other Notifications

- kEventControlHit
  - Sent after a click on a control
- kEventCommandProcess
  - Sent if the control has a command associated with it: SetControlCommandID()

DataBrowser



- List view
- Column view
- Customizable
- In-place editing
- Drag & Drop
- . . .and more!

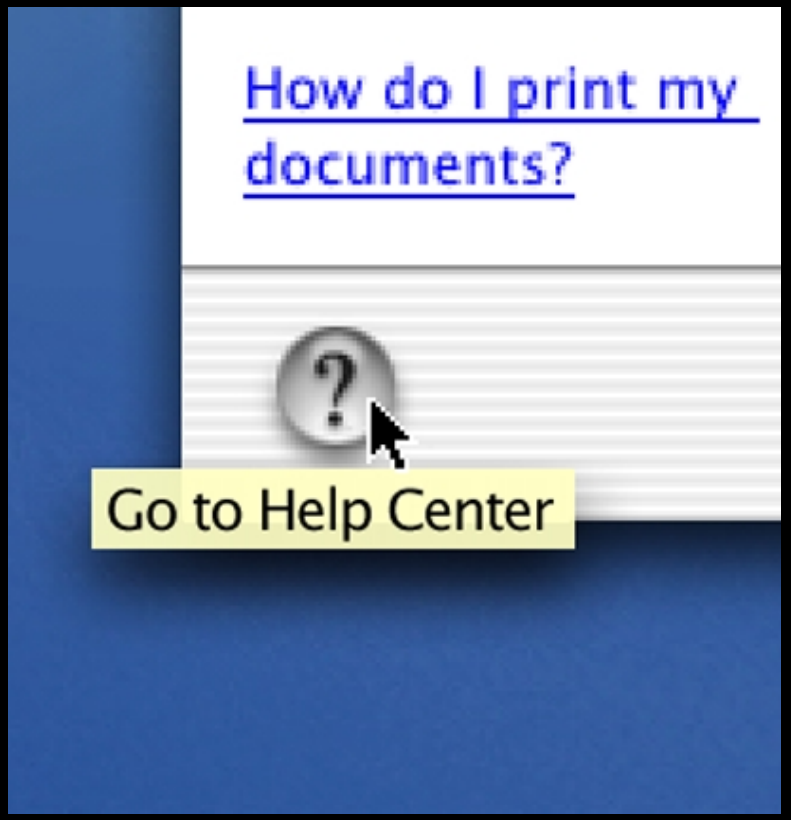
DataBrowser

- System-supplied Control Definition:
  - CreateDataBrowserControl()
- Use all the normal Control Manager APIs
- DataBrowser-specific APIs available in ControlDefinitions.h

DataBrowser Future

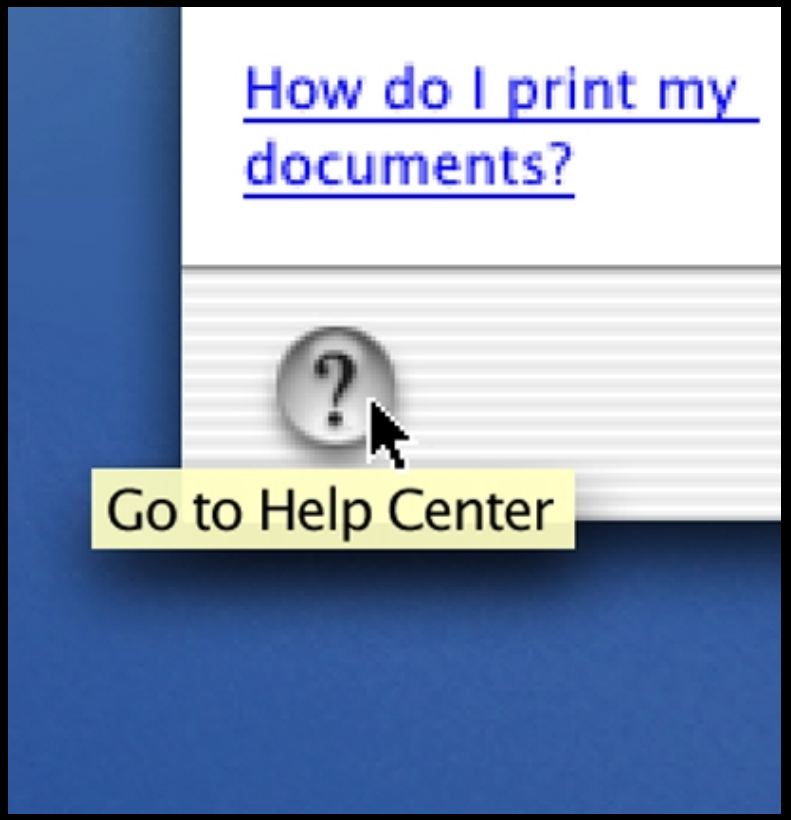
- Speed
- Bug fixes
- More speed
- CarbonEvent integration
- Features

Help Button



- CreateRoundButtonControl() on Mac OS X
- CreateBevelButtonControl() on CarbonLib
- Use Icon Services for the help icon:
  - GetIconRef()

Help Tags



- Context-sensitive tips
- Two-stage content
- Position customization
- Content guidelines available online

Implementing Help Tags

- Interface Builder .nib support
- Programmatic Interface
- Balloon Help content support via compatibility APIs

Implementing Help Tags

- Associate static help data with toolbox object:
  - HMSetControlHelpContent()
- Dynamic content via callbacks:
  - HMInstallControlContentCallback()
- Manual Display:
  - HMDisplayTag()

Drag & Drop

- New look for drag highlight
  - Draw with the proper theme brush SetThemeBackground() and kThemeBrushPrimaryHighlightColor
- Drag image can be changed on the fly
  - SetDragImage() instead of DrawProcs
- Rest of the APIs are the same

Cursor Setting

- Repetitive cursor changing hurts performance
- Wait Cursor Methodology
  - < 2 seconds: Display a busy cursor
  - > 2 seconds: System will displays a busy cursor
  - > 10 seconds: Use a modeless progress indicator

Ship It!

- Carbon Events are the future
- Take advantage of the Dock
- Make your application stand out!

Roadmap

119 Carbon Controls & Appearance I	Hall 2 **Thurs., 9:00am**
130 Designing & Using Aqua Icons	Hall 2 **Fri., 10:30am**
125 Apple Help	Room J2 **Thurs., 2:00pm**
127 Speech Recognition & Synthesis In-Depth	Room C **Thurs., 10:30am**
114 Application Packaging & Document Binding	Hall 2 **Wed., 5:00pm**

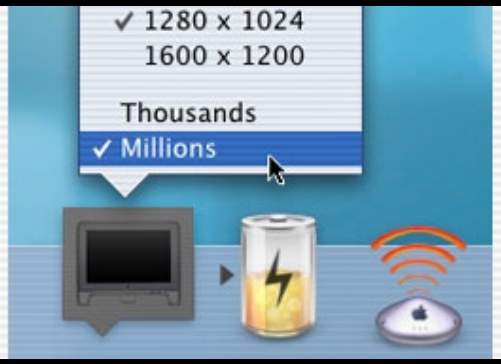
Roadmap

702 Interface Builder	Hall 2 **Tues., 3:30pm**
FF103 Feedback Forum: Aqua	Room J1 **Wed., 3:30pm**
FF114 Feedback Forum: High Level Toolbox	Room C **Fri., 3:30pm**
106 Carbon Event Manager	Hall 2 **Tues., 5:00pm**
112 Carbon: Windows & Menus	Hall 2 **Wed., 2:00pm**

Who to Contact

John Geleynse
User Experience Technology Manager
Apple Worldwide Developer Relations
geleynse@apple.com

Q&A



John Geleynse
User Experience Technology Manager
geleynse@apple.com

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001